

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-212929

(43)Date of publication of application : 06.08.1999

(51)Int.Cl.

G06F 15/16

(21)Application number : 10-014300

(71)Applicant : NEC COMMUN SYST LTD

(22)Date of filing : 27.01.1998

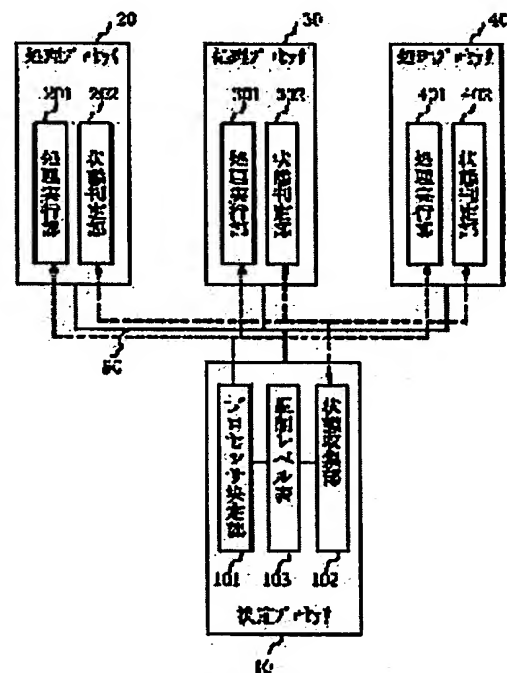
(72)Inventor : KIRIHARA SHIGEMITSU

## (54) PROCESSOR DECISION SYSTEM USING LOAD DISTRIBUTION MULTIPROCESSOR SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To effectively utilize the load-distributed processors.

SOLUTION: In this processor decision system, a decision processor 10 selects a processor to execute the processing if a processor 20 has high load among the processors 20, 30 and 40. In such cases, the processor 10 lowers the selection rate of the processor 20. Thereafter, the selection rate of the processor 20 is set at a level equivalent to those of other processors after the processor 10 detects that the processor 20 has the normal load again.



## LEGAL STATUS

[Date of request for examination]

27.01.1998.

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3394437

[Date of registration]

31.01.2003

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-212929

(43) 公開日 平成11年(1999) 8月6日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 15/16

識別記号

3 7 0

F I

G 0 6 F 15/16

3 7 0 N

審査請求 有 請求項の数 4 O L (全 6 頁)

(21) 出願番号 特願平10-14300

(22) 出願日 平成10年(1998) 1月27日

(71) 出願人 000232254

日本電気通信システム株式会社  
東京都港区三田 1 丁目 4 番 28 号

(72) 発明者 桐原 恵満

東京都港区三田 一丁目 4 番 28 号 日本電気  
通信システム株式会社内

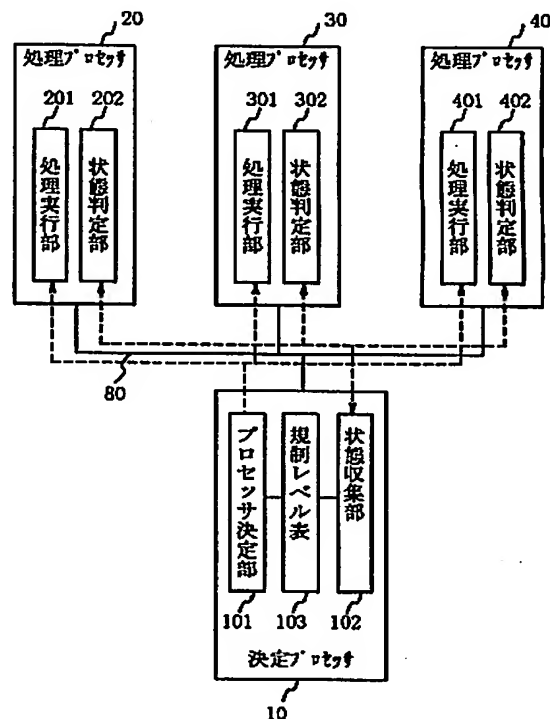
(74) 代理人 弁理士 京本 直樹 (外 2 名)

(54) 【発明の名称】 負荷分散マルチプロセッサ方式によるプロセッサ決定方式

(57) 【要約】

【課題】 負荷分散されたプロセッサの有効活用を実現することを目的とする。

【解決手段】 負荷分散されている処理プロセッサ 20, 30, , および 40 の内、プロセッサ 20 の負荷が高くなった場合、決定プロセッサ 10 は処理を実行させるプロセッサを選択する際、処理プロセッサ 20 に対する選択の割合を削減する。その後、決定プロセッサ 10 が処理プロセッサ 20 の負荷が正常に戻ったことを検出すると、処理プロセッサ 20 の選択の割合を他と同等なものとする。



## 【 特許請求の範囲】

【 請求項1 】 負荷分散された複数の第1 のプロセッサと、負荷分散された前記第1 のプロセッサの1 つを選択する第2 のプロセッサとを有する負荷分散マルチプロセッサ方式によるプロセッサ決定方式において、タスクの処理要求があった場合に前記第2 のプロセッサは、タスクの処理を実行させるために負荷分散された前記第1 のプロセッサの負荷に応じて複数の前記第1 のプロセッサの選択の判断を行うことを特徴とする負荷分散マルチプロセッサ方式によるプロセッサ決定方式。

【 請求項2 】 負荷分散された複数の第1 のプロセッサと、負荷分散された前記第1 のプロセッサの1 つを選択する第2 のプロセッサとを有する負荷分散マルチプロセッサ方式によるプロセッサ決定方式において、タスクの処理要求があった場合に前記第2 のプロセッサは、選択可能なプロセッサの数に応じて要求された前記処理の実行を行わないことを特徴とする負荷分散マルチプロセッサ方式によるプロセッサ決定方式。

【 請求項3 】 負荷分散された複数の第1 のプロセッサと、負荷分散された前記第1 のプロセッサの1 つを選択する第2 のプロセッサとを有する負荷分散マルチプロセッサ方式によるプロセッサ決定方式において、前記第2 のプロセッサは、一定間隔毎に前記第1 のプロセッサのCPU使用率と予め設定された輻輳判断数とを読み出し比較する比較手段と、前記CPU使用率が前記輻輳判断数以上であれば前記第1 のプロセッサのタスクの処理できる制限数を示す処理実行可能数を段階的に減少させる減少手段と、前記CPU使用率が前記判断数未満であれば前記第1 のプロセッサのタスクの前記処理実行可能数を正常負荷状態で最大の処理できる数になるまで段階的に復帰させる復帰手段とを有することを特徴とする負荷分散マルチプロセッサ方式によるプロセッサ決定方式。

【 請求項4 】 前記第2 のプロセッサは、前記第1 のプロセッサ毎に前記輻輳判断数と、前記処理実行判断数と、前記最大の処理できる数とを含む管理テーブルを備えることを特徴とする請求項3 記載の負荷分散マルチプロセッサ方式によるプロセッサ決定方式。

## 【 発明の詳細な説明】

## 【 0 0 0 1 】

【 発明の属する技術分野】 本発明は、負荷分散されたマルチプロセッサの起動方法に関し、特に過負荷時の負荷分散マルチプロセッサ方式によるプロセッサ決定方式に関する。

## 【 0 0 0 2 】

【 従来の技術】 従来、この種の負荷分散マルチプロセッサ方式によるプロセッサ決定方式は、例えば、特開平4 -2 3 5 6 6 2 号公報に開示されている。この従来の方式は負荷分散されたマルチプロセッサ方式のシステムに於いては、処理を実行するプロセッサを何らかの方法で決定する必要があり、更に各プロセッサには均等に処理

の要求が加えられることが求められる。しかし各プロセッサに要求される処理の内容は同一の物ではなく各プロセッサにかかる負荷には差が生じてくる。この結果特定のプロセッサの負荷が高くなることもある。負荷が高くなったプロセッサに対して更に処理を実行させるプロセッサの持つ処理能力を超えた負荷がかかり、プロセッサがデッドロックする事がある。これを防止するためにある一定以上の負荷がかかっているプロセッサには処理を起動せず、他のプロセッサへ処理を移す制御を行っている。

## 【 0 0 0 3 】

【 発明が解決しようとする課題】 上述のように、従来技術の方式では、一定以上の負荷がかかったプロセッサを検出すると該当プロセッサに対して一定時間処理を起動しなくなるため、処理要求を受けなくなったプロセッサの処理能力は急激に回復していた。更に、各プロセッサに対してかかる負荷の検出は、常時行われてはおらず、周期的な検出にとどまっていた。この両者の処理の結果、一度負荷が上昇し起動がかからなくなったプロセッサは、高負荷状態から正常負荷状態へ移行し処理を実行できるレベルになっても負荷の状態を見に行くまでの間、実際には処理が可能であるにもかかわらず、処理の要求を受けないと言う現象が発生するという問題点がある。

【 0 0 0 4 】 また、システムにかかる処理量が進むと、処理可能な能力の限界を越え起動を受けなくなるプロセッサの数が増加し、処理要求を受け付ける事ができるプロセッサの数が減少していく。この後更に負荷が上がると、少ない数のプロセッサに処理要求が集中し、一気に処理不可能な状態に陥ってしまい処理を受け付けるプロセッサが無くなってしまう。更にこの現象は周期的な負荷の検出によってのみ発見可能であるため、検出のタイミングが遅れると、全プロセッサがデッドロックとなる可能性があるという問題点がある。

【 0 0 0 5 】 本発明の目的は、負荷分散されたプロセッサの有効利用を提供することにある。

【 0 0 0 6 】 更に、本発明の目的は、プロセッサに対して処理限界を超えた処理要求が加わることにより発生するプロセッサのデッドロックを防止することにある。

## 【 0 0 0 7 】

【 課題を解決するための手段】 上記目的を達成するために、本発明は、負荷分散された複数の第1 のプロセッサと、負荷分散された前記第1 のプロセッサの1 つを選択する第2 のプロセッサとを有する負荷分散マルチプロセッサ方式によるプロセッサ決定方式において、タスクの処理要求があった場合に前記第2 のプロセッサは、タスクの処理を実行させるために負荷分散された前記第1 のプロセッサの負荷に応じて複数の前記第1 のプロセッサの選択の判断を行うことを特徴としている。

【 0 0 0 8 】 また、本発明は、負荷分散された複数の第

10

20

30

40

50

1のプロセッサと、負荷分散された前記第1のプロセッサの1つを選択する第2のプロセッサとを有する負荷分散マルチプロセッサ方式によるプロセッサ決定方式において、タスクの処理要求があった場合に前記第2のプロセッサは、選択可能なプロセッサの数に応じて要求された前記処理の実行を行わないことを特徴としている。

【0009】また、本発明は、負荷分散された複数の第1のプロセッサと、負荷分散された前記第1のプロセッサの1つを選択する第2のプロセッサとを有する負荷分散マルチプロセッサ方式によるプロセッサ決定方式において、前記第2のプロセッサは、一定間隔毎に前記第1のプロセッサのCPU使用率と予め設定された輻輳判断数とを読み出し比較する比較手段と、前記CPU使用率が前記輻輳判断数以上であれば前記第1のプロセッサのタスクの処理できる制限数を示す処理実行可能数を段階的に減少させる減少手段と、前記CPU使用率が前記判断数未満であれば前記第1のプロセッサのタスクの前記処理実行可能数をCPU正常負荷状態で最大の処理できる数になるまで段階的に復帰させる復帰手段とを有することを特徴としている。

【0010】更に、前記第2のプロセッサは、前記第1のプロセッサ毎に前記輻輳判断数と、前記処理実行判断数と、前記最大の処理できる数とを含む管理テーブルを備えることを特徴としている。

【0011】

【発明の実施の形態】図1は本発明の実施の形態におけるブロック構成図を示す。

【0012】図1を参照すると、プログラム制御により動作する決定プロセッサ10は、プログラム制御により動作する複数の処理プロセッサ20、30、40とバス80で接続されている。

【0013】決定プロセッサ10は負荷分散された処理プロセッサ20、30、40を選択および起動する為のプロセッサである。処理プロセッサ20、30、40は実際の処理を行うプロセッサで、前述の決定プロセッサ10から起動命令を受け付ける。ここでの処理プロセッサ20、30、40は負荷分散されていることを示すために複数記述してあるものであり、各プロセッサの処理内容は全て同一である。

【0014】決定プロセッサ10は、処理を実行させるプッサを決定する決定部101と、判定を行う際に判断材料として使用する各処理プロセッサのCPU使用率を収集する状態収集部102と、状態収集部102で収集したプロセッサのCPU使用率が許容範囲を超えていた場合の処理プロセッサ起動回数の増減を決定する処理規制レベル表103とから構成される。それぞれの処理プロセッサ20、30、40は、決定プロセッサ10内の状態収集部102が収集する処理プロセッサ20、30、40のCPU使用率を判断する状態判断部202、302、402と、システムの本来の目的のための処理を行

う処理実行部201、301、401とから構成される。

【0015】処理規制レベル表103のフォーマットは、各処理プロセッサ毎に輻輳状態(後述説明)のレベルの変数を示す「n」と、輻輳状態から段階的に最大処理できる状態に戻すための変数を示す「m」、処理プロセッサの処理できる最大の数を示す「処理実行最大数」と、処理プロセッサが処理できる実行の数量を示す「処理実行可能数」と、輻輳状態と判断するCPUの使用率を示す「輻輳判断数」とを含んでいる。「n」はシステム立ち上げ時には値「0」が設定される。「処理実行最大数」、「処理実行可能数」、「輻輳判断数」、および「m」はシステムの立ち上げ時に各プロセッサの能力に応じて初期値が設定される。

【0016】次に、図1のブロック構成図、および図2のプロセッサ決定時のフローチャートを用いて実施の形態の一例の動作の説明を行う。

【0017】図1において、処理プロセッサ20、30、40は負荷分散されたプロセッサで同一目的の処理を行うために設置される。この場合、処理の要求が発生した時点でどの処理プロセッサに処理を実行させるかを決定する必要があるが、この判断を行うのがプロセッサ決定部101である。プロセッサ決定部101は基本的には各処理プロセッサを同じ割合で起動するが、特定の処理プロセッサのCPU使用率が何らかの要因で上昇(以下輻輳状態と言う)しこれ以上起動をかける事が出来なくなった場合選択の対象から除外する必要がある。この時、処理プロセッサのCPU使用率の監視は状態収集部102が周期的に状態判定部202、302、402に問い合わせることで実現する。

【0018】状態収集部102は全処理プロセッサの輻輳状態を周期的に把握し、輻輳状態と判断されると、該当プロセッサの前状態から規制のレベルを判断し処理規制レベル表103に記憶させる。このときのCPU使用率の収集は以下の通りである。

【0019】各処理プロセッサにおいて、CPUに対しての随時発生する処理要求は周期(例えば8ms)毎に割込をかけ、ドライバ(制御プログラム)によって待ち行列(処理を行うために待っているタスク)から取り出す。個々での処理要求を全て処理終了するのに要した時間の8msに占める割合が瞬間的なCPU使用率となる。状態判定部202、302、402は、この8ms単位のCPU使用率を随時測定および加算している。このとき、周期的(例えば20秒間隔)な状態収集部102からの要求によりそれまでの20秒間の平均を算出(20秒間に占める割合)し収集した値をCPU使用率として状態収集部102に転送する。

【0020】決定プロセッサ10の状態収集部102が動作する場合の処理フローを図2に示す。

【0021】図2において、20ms毎に割込がかかる

と、状態収集部102は、最初の処理プロセッサを選択する。更に、処理プロセッサの状態判定部からCPU使用率を読み出すと共に、処理規制レベル表103の「輻輳判断数」を読み出し、CPU使用率と「輻輳判断数」との比較を行う(ステップS1, S2)。

【0022】CPU使用率が「輻輳判断数」以上であれば、輻輳状態と判断し「処理実行可能数」(yとする)を処理規制レベル表103から読み出し、値が「0」であるかどうかチェックする。値が「0」でなければステップS5のような計算を行う。なお、式 $y = z * (0.6 - 0.1 * n + 0.2 * m)$ は、タスクの処理できる実行数を輻輳状態に応じて段階的に決めるための一例となる式であって、必ずしもこの式にする必要がない。式中のzは処理規制レベル表103の「処理実行最大数」を示す。例の処理では、「処理実行可能数」は最初「処理実行最大数」の50%にし、それ以降は徐々に10%ずつ下げるようにしている(ステップS4, S5)。

【0023】CPU使用率が「輻輳判断数」未満であれば、正常負荷状態と判断し処理規制レベル表103の「m」を読み出し、値が「0」かどうかをチェックする。m=0であればステップS7のような処理を行う。なお、式 $y = z * (0.6 - 0.1 * n + 0.2 * m)$ は、輻輳状態から最大処理できる状態に戻すためのタスクの処理できる実行数を段階的に決めるための一例となる式であって、必ずしもこの式にする必要がない。例の処理では、最大処理できる状態に戻すために、「処理実行可能数」を「処理実行最大数」の20%ずつ上げるようにしている(ステップS6, S7)。なお、最大処理できる状態とは、処理規制レベル表103の「処理実行可能数」が「処理実行最大数」の値になったことをいう、すなわち各処理プロセッサのタスクの処理が最大数まで処理できる状態をいう。

【0024】m=0でなければ、処理実行可能数と処理実行最大数(完全に復帰したかどうかの確認)とを比較し、一致したらステップS7に進む(ステップS8, S9)。

【0025】処理プロセッサの指定はローテーションしながら順番(処理プロセッサ20→30→40)に行っていくが、次のプロセッサがまだ存在すればステップS2に飛ぶ。なければ終了する(ステップS10)。

【0026】なお、20ms毎に1ローテーションの処理を行う。

【0027】図4において、輻輳状態となった処理プロセッサ20に対する処理要求の規制レベルの管理方法例を示す。縦軸は単位時間あたりに処理要求を規制するレベルを示す。横軸に時間の経過を示す。このときの「処理実行最大数」は100、「輻輳判断数」は50%にシステム立ち上げ時に設定されている。

【0028】状態収集部102は周期的に各処理プロセッサの状態判定部202, 302, 402よりCPU使用率を収集する。

【0029】このとき処理プロセッサ20の状態判定部202からCPU使用率を収集した状態収集部102は基準値を超えたCPU使用率を検出した場合、該当プロセッサを輻輳状態と判断し処理実行可能数の規制を実施する。処理実行可能数の規制は、それまでのCPU使用率が正常であった場合、処理実行可能数を大幅に減少させ(図2のステップS5によれば50%ダウン)、1度輻輳状態となった処理プロセッサが次の収集周期においても輻輳状態が復旧していない場合、処理実行可能数を小幅(図2のステップS5によれば10%ずつダウン)に減少させる。この小幅な処理実行可能数の減少処理は、輻輳状態となった処理プロセッサが最大処理できる状態になるまで続けられる(図2のステップS7によれば、正常負荷状態であると20%ずつ回復させる)。このようにして、プロセッサ決定部101は処理を実行させるプロセッサを決定する際に処理規制レベル表103を参照し処理実行可能状態かの判断を実行する。

【0030】次に、タスク処理の要求があった場合、プロセッサ決定部101が処理プロセッサを決定する動作例を図3に示す。

【0031】図3において、処理プロセッサを順番に処理実行候補プロセッサとして選択する(ステップA1)。候補として選択した処理プロセッサが輻輳状態にあるかを規制レベル表103から判断する(ステップA3)。ここで選択した処理プロセッサが輻輳状態でなければこの処理プロセッサを処理実行対象プロセッサとして決定する。(ステップA3)。輻輳中であれば処理実行可能数の残り回数を判断し(ステップ4)、処理要求実施可能範囲内(処理プロセッサの許容範囲内)であれば処理実行プロセッサとして選択する(ステップA5)。更に、その場合規制レベルデータの処理実行可能数を1減算する。処理実行可能数が0になっていた場合、他の処理プロセッサを処理実行候補プロセッサとして選択し(ステップA6)、ステップA2へ戻り処理可能かの判断を行う。

【0032】以上説明したように、各処理プロセッサの処理実行可能数の制限をCPU使用率に基づいて決めているため、処理プロセッサの処理能力を無駄にすることなく運用できる。

【0033】上述した実施の形態の一例に於いては、複数のプロセッサが輻輳状態となり処理対象から除外されてしまった場合、残った少数のプロセッサへ急激に処理が集中してしまい、集中したプロセッサが前述した発明の効果をj得る時間の余裕をなくし要求された処理量が処理能力を一気にオーバーしてしまうため、最悪処理プロセッサがデットロックの状態になる可能性がある。このため、輻輳中プロセッサ及び、正常運転中プロセッサの数を管理し、正常運転中のプロセッサの数が一定数を下回った場合、要求のあった処理の実行を行わないようにす

る。このとき処理規制制御レベル表103に「処理可能プロセッサ数」を設定すればよい。この「処理可能プロセッサ数」はシステム立ち上げ時に初期設定され、処理不可能(「処理実行可能数」=0)になったプロセッサが生じた場合は「1」減らすようにする。この場合の決定プロセッサ10が処理プロセッサを決定する場合の動作例を図5に示す。

【0034】図5において、ステップA1～A7については図3での説明と同じなので説明を省略する。図5のステップ5において、受付許容数が範囲外であれば「処理可能プロセッサ数」の値をチェックし、値が「0」であれば処理要求を破棄する(ステップA11、A12)。「処理可能プロセッサ数」の値が「0」より大きい場合は、次の処理プロセッサを処理候補プロセッサとする(ステップA6)。

【0035】上記説明したように、負荷状態に応じて処理可能プロセッサの数に制限を加えることで、急激な負荷にも耐えることができ、システムの安定運用ができる。

【0036】

【発明の効果】以上説明したように、負荷状態に応じてプロセッサの処理の実行できる数の制限を行っているため、負荷分散されたプロセッサの内いずれかが輻輳状態となっても、該当するプロセッサの処理能力を無駄にすることなく運用することが出来るという効果がある。

【0037】更に、負荷状態に応じて処理可能プロセッサの数に制限を加えるようにしているため、プロセッサ

が負荷分散されたシステムに対して、急激な負荷の上昇があった場合でも、システムの安定状態を保つことが出来るという効果がある。

【図面の簡単な説明】

【図1】本発明の実施の形態の一例に於ける、プロセッサが負荷分散されたシステムのブロック構成を示す図である。

【図2】図1の各処理プロセッサの状態収集部に於ける、タスクの処理のできる制限数の設定に関するフローチャートである。

【図3】図1の決定プロセッサ10の状態収集部に於ける、タスクの処理のできるプロセッサの決定に関するフローチャートである。

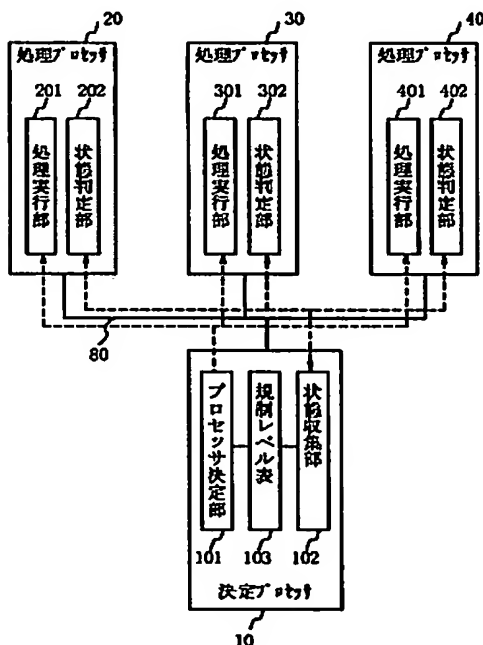
【図4】本発明の実施の形態の一例に於ける、ある特定の処理プロセッサに対する処理要求の段階的規制状況を表す図である。

【図5】本発明の実施の形態の一例における、負荷分散されたプロセッサの選択決定に関するフローチャートである。

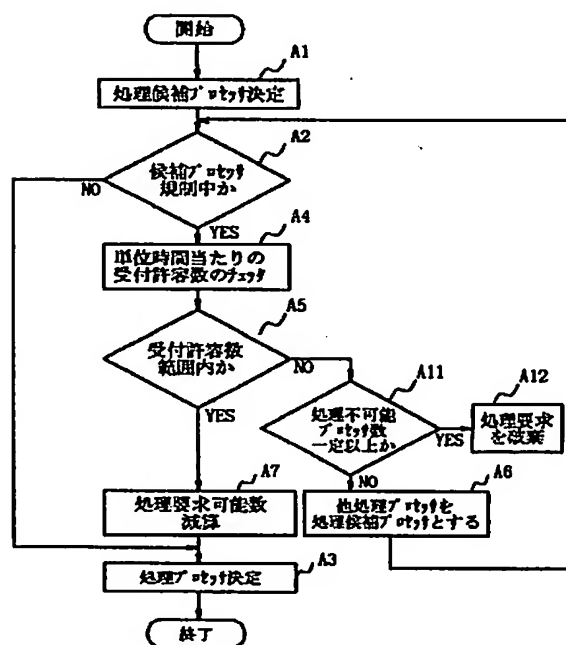
【符号の説明】

- 10 決定プロセッサ
- 20, 30, 40 処理プロセッサ
- 101 プロセッサ決定部
- 102 状態収集部
- 103 規制レベル表
- 201, 301, 401 処理実行部
- 202, 302, 402 状態判定部

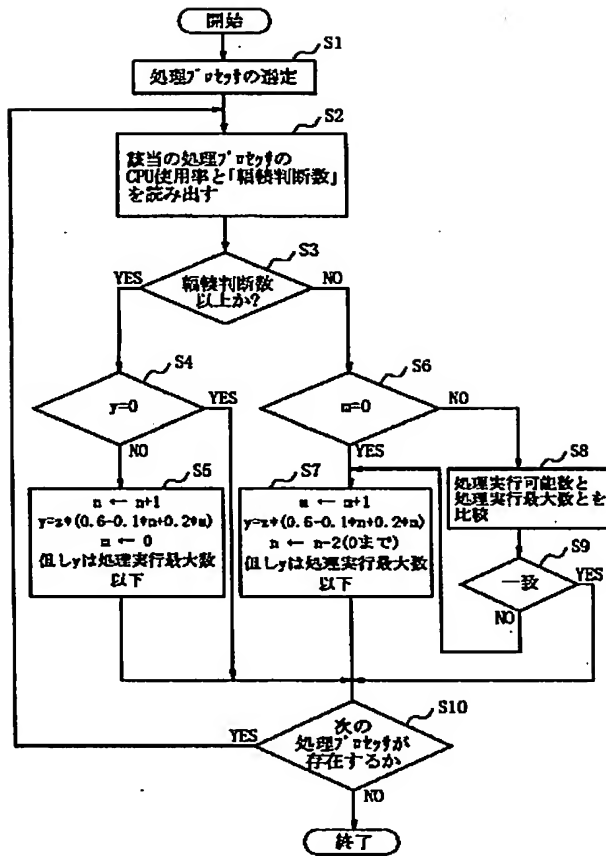
【図1】



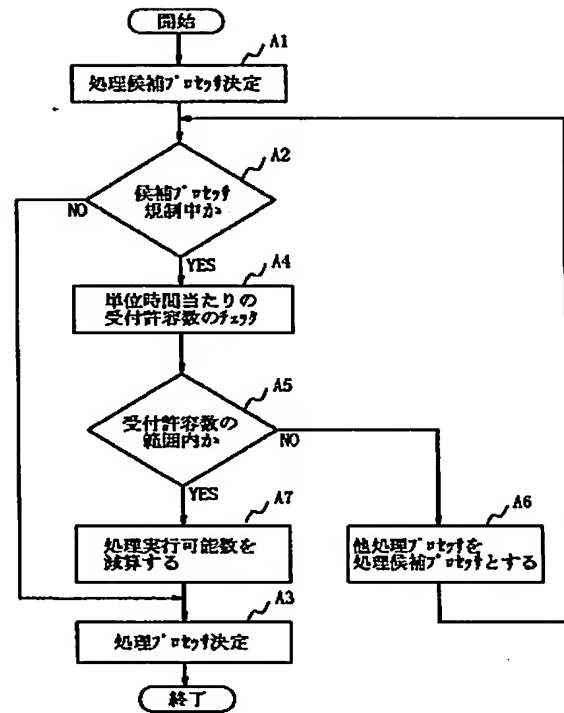
【図5】



【 図2 】



【 図3 】



【 図4 】

